



## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**APPLICANT:** Sprogis **GROUP:** 3602  
**SERIAL NO:** 09/627,870 **EXAMINER:** Gravini, S.  
**FILED:** July 28, 2000 #17  
**FOR:** SYSTEM AND METHOD FOR DIGITALLY  
PROVIDING AND DISPLAYING ADVERTISEMENT  
INFORMATION TO CINEMAS AND THEATRES

**Box AF**  
**Assistant Commissioner of Patents**  
**Washington, D.C. 20231**

## AFFIDAVIT OF DAN U. ZEHME

1. I, Dan Zehme, of 12 Newport Lane, Medfield, Massachusetts 02052, hereby declare as follows:
2. I am the Chief Technical Officer for CineCast LLC, a Delaware Limited Liability Corporation located at 121 Columbia Street, Cambridge, MA 02139, and have been employed by CineCast LLC since January 3, 2000.
3. In 1989 I received a Bachelor of Science degree from Rensselaer Polytechnic Institute in Computer and Systems Engineering. From 1989 to the present I have worked with the design, development and implementation of computer database systems. A copy of my resume is attached hereto as Appendix A.
4. I have reviewed the disclosure in U.S. Provisional Patent Application Ser. No. 60/148,807 filed August 13, 1999 (a copy of which is attached hereto as Appendix B). I

first saw a copy of this disclosure on or about January 3, 2000. I have also reviewed the disclosure in U.S. Patent Application Ser. No. 09/627,870 filed July 28, 2000 (a copy of which is attached hereto as Appendix C). I first saw a copy of this disclosure on or about July 28, 2000.

5. I believe that the disclosures in each of the applications that are attached as Appendices B and C include sufficient information for someone with my background and experience to build a system as described therein.

6. In particular, the data that is selected as an example in the patent application responsive to movie schedule information is data that relates to jobs. For example, the patent application states that:

The content provider creates a job that is stored in the database 70 by way of similar PHP script instruction(s) and transactional service(s). Among the information stored as part of the job is a requested schedule of showings for the content, the location of the showings and the content itself.

Application, p.22, line 21 - p.23, line 1. The patent application further states that:

A job artifact is the combination of message content and its context. The content could be, for example, a static or animated GIF, a Shockwave segment or anything else that can be displayed by or hosted in HTML or XML. The context is a schedule request, target audience, and other pertinent information. While a job object (job) is a transient object with a finite start, and end, it has a continuing representation in the table CC\_JOB and its supporting tables.

Application, p.30, lines 6 - 11.

This tells me an example of what a job may be and that the jobs may be stored in the table named CC\_JOB. This description is also provided at page 12, paragraph 3 and page 16, paragraph 6 of the provisional patent application.

7. The movie schedule information that is disclosed in the patent application as an example and responsive to which data (e.g., job data) is selected, is movie showing data.

For example, the patent application states that:

A showing object is a particular movie shown on a particular screen at a particular time. A schedule object is created by the schedule daemon to reserve a job for a particular showing, and these objects are represented in CC\_SHOWING and CC\_JOB\_SCHEDULE.

Application, p.33, lines 1 - 4. The patent application further states that:

A movie is a particular attraction. The movie has attributes such as genre and rating that among other attributes will naturally appeal to a particular demographic of moviegoer. A screen is a movie screen which is at a physical site (or location) 266 and which belongs to an exhibitor circuit 268. A screen has particular attributes including geographical location and dimensions and seat/sound enhancements which will naturally attract a particular demographic of moviegoer.

Time is also an important aspect of a showing. Because time is linear, it may be referenced using standard numeric notation and maintained using standard time/date persistence techniques. A showing 270 is the intersection of exactly one movie at one time on one screen. A showing is typically established 7 - 14 days in advance. A receipt 272 is a record of attendance of a particular show.

Application, p.35, lines 3 - 13.

This tells me an example of what the movie schedule data may be and that this data may be stored in the table named CC\_SHOWINGS. This description is also provided at page 18, paragraph 6; and page 3, paragraph 1 of the provisional patent application.

8. An example of the selection of data (e.g., job data) responsive to movie schedule information is fully disclosed in the patent application. For example, the patent application describes the schedule daemon as follows:

The schedule daemon is a continuously running process that connects contents from jobs to individual showings of movies. There is a limited number of contents that may be contained in a presentation, and this number is read and calculated from the designated presentation format and frame records. The schedule daemon reads the showings (CC\_SHOWING) as they become available on the system. Ordered by approval date, schedule requests (CC\_SCHEDULE\_REQUEST) of the job are read, and entries are added to CC\_JOB\_SCHEDULE. After the presentation's capacity for contents has been filled all other requests are left unsatisfied. If a job request cannot be satisfied, then a notice in the form of an email or login message or part of the billing report is conveyed to the content provider. However, capacity may become available as new showings are added or previously scheduled jobs are canceled.

Application, p.37 lines 3 - 13. The patent application further describes a job as:

A job may have one or more schedule requests defining where and when the content will be displayed. However, it is only a request and might not be fulfilled. Request fulfillment is the responsibility of the schedule daemon. As shown in Figure 9, the schedule request has up to four dimensions describing when and where it will be shown, start and end dates, times of day, location, and movie.

Application, p.30, lines 12 - 16. The process of automatically selecting jobs responsive to movie scheduling information is specifically detailed as an example in the patent application, which states that:

A schedule request 258 details the delivery of the jobs content to an intersection of one or more of time 260, movie 262, and or screen (location) 264. Therefore, a request may be as general as placing an ad against a particular movie or as specific as a particular movie at a particular time and in a particular location. With multiple requests per job, an advertiser may reach multiple market segments with the same job content. A job schedule is a job request that has been resolved to a particular show. Because shows, themselves, are scheduled 7-14 days in advance and because advertisers and agencies request shows weeks and months in advance, a job request can go unresolved for some period of time.

Application, p.34 lines 17 - p.35, line 2. The patent application further states that:

As shown in Figure 12, there exists a variety of permutations of schedule requests based on different combinations of factors such as the location 300, the movie 302, the time of day 304, and the date 306. The area indicated at 300 represents schedule request where the location is the specifically requested location, and no other criteria are specified. The area indicated at 302 represents schedule request where the movie is the specifically requested movie, and no other criteria are specified. The area indicated at 304 represents schedule request where the time of day is the specifically requested time of day, and no other criteria are specified. The area indicated at 306 represents schedule request where the date is the specifically requested date, and no other criteria are specified.

The overlapped areas represent combinations of specified criteria as shown. For example, the area indicated at 308 represents a combination of specifically requested location and time-of-day factors, where no other criteria are specified. The area indicated at 310 represents a combination of each of the location, movie, time of day, and date criteria.

These permutations translate directly into one or more SQL queries used to build the list. Once the list is built, duplicates are removed as it is currently assumed that duplicates are undesirable. Duplicates may be present because the schedule requests overlap, not because the queries overlap. This list is used to populate CC\_JOB\_SCHEDULE with schedules.

Application, p.37, line 18 - p.38, line 14.

This tells me that the job data may be selected responsive to movie scheduling data (e.g., location, movie, time of day, date etc.) in accordance with conventional relational database standards that were in existence in 1999. For example, a conventional Structured Query Language (SQL) query involves the following format: SELECT <attribute\_list> FROM <table(s)> WHERE <condition=desired>. This elemental programming step was widely known and utilized in 1999, particularly for joining dissimilar entities with common attributes to yield a linked result. This description is also provided at page 21, paragraph 1 - page 22, paragraph 1; and page 17, paragraph 2 of the provisional patent application.

9. An example of providing of a first portion of data representative of advertisement information to a first digital projector assembly responsive to first theatre scheduling information regarding a movie that is to be shown in the first theatre is fully disclosed in the patent application and in the provisional application as discussed above in paragraphs 6 - 8. In particular the first portion of data representative of advertisement information is disclosed, as an example, to include job data. The first theatre scheduling information is disclosed, as an example, to include the movie showing data. The selection is achieved as discussed in paragraph 8 above as an example using conventional SQL programming steps and the tables CC\_SHOWINGS and CC\_JOBS.

10. An example of movie identification information regarding a movie that is to be shown in a theatre environment associated with a first digital projector assembly, and responsive to which data is selected from the computer storage unit, is movie information data. For example, the patent application states that:

The movie object is a description of the film's content. A release object is an extension of the movie object indicating release date, expected gross, and other studio estimates. This relationship facilitates re-releases, limited releases, and sneak previews. Movies and releases will be populated as they become known from the studios. Second run movies and classic movies will be back-filled. Movies and releases are represented in CC\_MOVIE, CC\_MOVIE\_RELEASE and supporting tables.

Application, p.32, lines 18 - 23. The patent application further states that:

Movie information is stored in the movie and release database along with a description, genera classification, expected box-office gross and other pertinent information.

Application, p.25, line 21 - p.26, line 1.

This tells me what the movie information data may be and that this data may be stored in the tables named CC\_MOVIE and CC\_MOVIE\_RELEASE. This description is also provided at page 18, paragraph 5 and page 14, paragraph 2 of the provisional patent application.


11. An example of selecting certain stored data from the computer storage unit for transmission to a first digital projector assembly of the plurality of digital projector assemblies responsive to movie identification information regarding a movie that is to be shown in a theatre environment associated with the first digital projector assembly is disclosed in the patent application and in the provisional application as discussed above in paragraphs 6 and 8. In particular the stored data is disclosed, as an example, to include job data. The movie identification information is disclosed, as an example, to include the movie data as discussed above in paragraph 10. The selection is achieved as discussed in paragraph 8 above as an example using conventional SQL programming steps and the

tables CC\_MOVIE, CC\_MOVIE\_RELEASE and CC\_JOBS. Specifically, the disclosure tells me that the job data may be selected responsive to movie information (e.g., movie, rating, genre etc.) in accordance with conventional relational database standards that were in existence in 1999. Again, a conventional SQL query involves the following format: SELECT <attribute\_list> FROM <table(s)> WHERE <condition=desired>. This elemental programming step was widely known and utilized in 1999, particularly for joining dissimilar entities with common attributes to yield a linked result. This description is also provided at page 21, paragraph 1 - page 22, paragraph 1; and page 17, paragraph 2 of the provisional patent application.

12. I acknowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001, and that such willful false statements are the like may jeopardize the validity of the application or document or any registration resulting therefrom, and I further declare that all statements made of my knowledge are true and that all statements made on information and belief are believed to be true.

\*\*\*\*\*

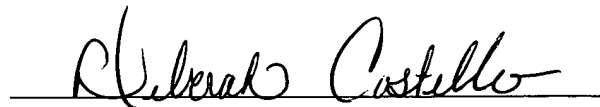
Date: March 25, 2003



Dan U. Zehme  
12 Newport Lane  
Medfield, Massachusetts 02052

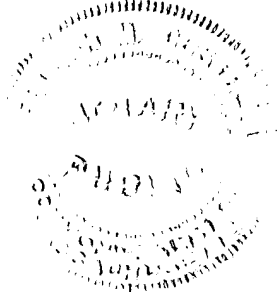
Notary:

Date: 3/25/03



Name: Deborah Costello  
My Commission Expires: 3/25/03

Seal





## DAN ZEHME

12 Newport Lane, Medfield MA 02052  
(508) 242-9992, dzehme@zehme.com

- OBJECTIVE** A position with a software development corporation where I can use my managerial and development skills developing Microsoft Windows or web applications.
- SKILLS**
- Java, JavaScript, C++, C, Pascal, BASIC, PowerScript, SQL, and assembly languages.
  - Web Technologies including JSP, SSL, HTML, DHTML, XML, and J2EE.
  - Microsoft Windows using SDK, MFC, or ATL.
  - Microsoft Active Automation, Document, Embedding, and Control technologies.
  - Windows NT, Windows, MS-DOS, Linux, UNIX.
  - Object oriented analysis and design methods.
  - Educated in hardware design.
- EXPERIENCE**
- 2000–2003                      CineCast, LLC                      Chestnut Hill, MA  
**Chief Technology Officer**
- Developed a system using commodity components for the management, delivery, and presentation of digital advertising in movie theaters as a replacement to existing slide advertising. The system provides increased revenues for the exhibitor, an improved experience for the moviegoer, and a current, targeted program for the advertiser.
  - Delivered presentations and participated in contract negotiations with General Cinemas, Loews, United Artists, NCN, Technicolor and others. Participated in business plan development, budgeting, and valuation analysis.
  - Created a web site for the management of digital advertising content using Apache, Secure Sockets (SSL), Tomcat JSP, Java, JDBC, and Sybase and MySQL Databases.
  - Created regional, site and screen server daemons in Java for the scheduling and delivery of digital advertising content, delivery of logs, delivery of errors, retrieval of movie listings, and extraction of show time and attendance data from exhibitor point-of-sale systems.
  - Created a presentation player using Visual Basic, Internet Explorer, Windows Media Player and C++, displaying feature, sponsor and advertising content on the screen in three frames with background and synchronized music.
  - Specified and tested an appliance PC for site and screen servers using PCs without keyboard or monitor, a custom mounting bracket, custom activation cable, digital projector, 100-BaseT LAN, and fractional-T1/DSL WAN.
  - Performed site surveys at theaters in the NYC area, collected requirements, and wrote change documents, user documentation, and functional specifications.
- 1994–1999                      Sybase, Inc.                      Concord, MA  
**Development Manager and Architect**
- Promoted to development manager within first two years.
  - Developer, manager, and architect for the integration of *PowerSite* (HTML) into the *PowerJ* (Java) and *PowerBuilder* (PowerScript) for *EASudio 4.0*. Work included defining and implementing interfaces allowing the maximum amount of code sharing with the least amount of changes to each of the development environments and refining the development of EJB components, application servers, and web sites.
  - Built and managed a 10 person User Interface team for *PowerBuilder* and *InfoMaker*, versions 5.0, 6.0, and 7.0. *PowerBuilder* is a PowerScript client/server and component, development environment written in C++ and the Windows API. Set the direction and overall architecture of the development environment as well as writing code and bug fixing.

- Redesigned the user interface for *PowerBuilder 7.0* using the principles of information presentation, program flow, and the transformation of existing code instead of rewriting. This included a new development methodology that organized and simplified the use of new features in the development environment and an innovative tiled and tabbed system for managing views. Worked hands-on writing several large portions of the new interface.
- Responsible for International development including eight European languages, Arabic enabled, Hebrew enabled, Kanji DBCS, and Unicode. This work included adding 32-bit BIDI and Unicode support to *PowerBuilder*.
- Built and managed a 6 person team for *NetImpact Studio* (later *PowerSite*), a HTML centric Internet development tool written in C++, MFC, and ATL. Responsibilities included setting product definition, task assignments, scheduling, and coding. Alerted management to serious problems in the execution of this project many months before targets.
- Designed and managed implementation of a tree based HTML manipulation engine with unlimited undo and the entire bookmark, site, and application management system for *PowerSite*.
- Implemented multi-platform versions of image list, tab, list view, tree view, header, tool tips, property sheets, wizards, and toolbars using C++ and the Windows API. Integrated Watcom's color editor.
- Developed build and source control procedures.
- Maintained the highest bug fix rate, often exceeding 50 issues per week.
- Presented for customer advisory board meetings, user conferences, and on-site training.

1989–1994

Data General Corporation

Westborough, MA

**Senior Software Engineer**

- Earned two promotions in three years to Senior Software Engineer.
- Enhanced *CEO Object Office*, a Microsoft Windows and Hewlett-Packard New Wave based office automation product. Implemented interface to Windows Sockets. Reorganized product build resulting in significant savings in development time. Eliminated many serious bugs in other workers' source code.
- Reimplemented *MV Server for Microsoft MS-NET (MVNET)*, a Microsoft Networking printer and file sharing server implemented using C and NetBIOS. Increased open file limits from 255 to 4000. Improved directory performance by more than 50%. Improved file performance by more than 20%. Simplified file time and attribute handling. Added a broad set of user customization features. Removed security deficiencies. Improved error reporting and added operation logging.
- Reworked *Distributed Configuration Server (DCS)*, a distributed NetBIOS name server implemented using C, SQL, and NetBIOS. Eliminated peer to peer time skew problems. Increased peer to peer connection reliability. Streamlined and simplified the user interface. Reduced database size by more than 50%.
- Managed development of the \$3 million/year *TermServer* terminal server product line. Developed utilities to fill shortcomings. Set development constraints. Coordinated with 3Com, contractors, support, quality assurance, marketing, documentation, and manufacturing. Prioritized schedules, directed resources, and conducted meetings.

1988

Perkin-Elmer

Norwalk, CT

**Cooperative Education**

- Designed and developed *LC-235 View*, and application performing automatic upload, download, display, and analysis of ultraviolet spectra. It is implemented in C and Digital Research's GEM. This project began as a give-a-way but became a full-fledged product.

**EDUCATION**

1989

Rensselaer Polytechnic Institute

Troy, NY

B.S., Computer and Systems Engineering